

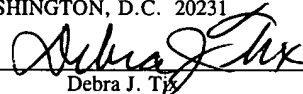
**PATENT**  
**5181-38400**  
**P4359**

"EXPRESS MAIL" MAILING LABEL

NUMBER EL487082797US

DATE OF DEPOSIT January 19, 2000

I HEREBY CERTIFY THAT THIS PAPER OR  
FEE IS BEING DEPOSITED WITH THE  
UNITED STATES POSTAL SERVICE  
"EXPRESS MAIL POST OFFICE TO  
ADDRESSEE" SERVICE UNDER 37 C.F.R.  
§1.10 ON THE DATE INDICATED ABOVE  
AND IS ADDRESSED TO THE ASSISTANT  
COMMISSIONER FOR PATENTS,  
WASHINGTON, D.C. 20231

  
Debra J. Tighe

System and Method for Enhancing Communication  
Between Devices in a Computer System

By:

David A. Wood  
Robert C. Zak Jr.  
Monica Wong-Chan  
Christopher J. Jackson  
Thomas P. Webber  
Mark D. Hill

## **BACKGROUND OF THE INVENTION**

### 1. Field of the Invention

This invention relates generally to the field of computer systems and, more particularly, to communication protocols within computer systems.

### 2. Description of the Related Art

High speed, low latency communications networks that include unreliable transport media often rely on a communications protocol to implement a reliable message transport. Examples of such communications protocols include TCP, NGIO 1.0, and PCI 2.x. In some of these protocols, a request can be sent from a sending device to a target device and an acknowledgment (ACK) can be sent in response from the target device back to the sending device. The sending device may include a timeout mechanism such that it can resend the request if an ACK is not received from the target device within a timeout duration set by properties of the communications network.

Some protocols may use a negative acknowledgement (NAK) to indicate that the target device or the communications network has detected an error. Errors can include data corruption, an illegal packet type, etc. The NAK can give a positive indication that an error has occurred and may also indicate the type of error that occurred. A sending device may, depending on the communications protocol, resend the request in response to a NAK.

In some communications networks, certain types of errors may temporarily prevent a target device from processing an incoming request. These types of errors can include a temporary loss of system resources (e.g., a dynamic reconfiguration of a node), a temporary lack of processing resources on the target device, or a lack of a valid virtual to physical address translation in cases where the contents of the request are to be written in the virtual address space

of the target device's node. While these errors may be temporary, the time required to resolve them can vary widely. For example, a dynamic reconfiguration of system resources in a server may take on the order of hundreds of milliseconds to resolve, a page miss in the virtual memory system may take on the order of tens of milliseconds to resolve, and a temporary resource  
5 unavailability in the network interface may take on the order of hundreds of microseconds to resolve. Thus, the time that the temporary unavailable condition persists may vary by four orders of magnitude or more.

When a target device is temporarily unable to process a request, it can send a NAK to the  
10 sending device. The sending device can later resend the request, but it may again receive a NAK from the target device if the temporarily unavailable condition has not been cleared. This process could potentially repeat a large number of times and result in a large increase of traffic on the communications network. Alternatively, the sending device may delay the resending of the request too long (i.e. well beyond the time needed for the target device to resolve the  
15 temporarily unavailable condition). As a result, unnecessary latencies may result in the sending device as the processing of its request is delayed. A system and method is needed to more efficiently handle conditions where a target device may be temporarily unavailable.

## SUMMARY

The problems outlined above are in large part solved by the use of the apparatus and method described herein. Generally speaking, an apparatus and method for resending a request in a computer system using a delay value is provided. In response to receiving a request, a target device in a computer system may detect that it is temporarily unable to process the request. The  
25 target device can send a response to the sending device to indicate that it is temporarily unavailable. The response can include a delay value that can provide a hint to the sending device as to when to resend the request. The target device may generate the delay value according to the type of condition that is causing it to be temporarily unavailable. The delay value may be generated according to a static heuristic or a dynamic algorithm based on previous temporarily

unavailable conditions. The delay value may also be used by an error recovery mechanism where a sending device exceeds a retry limit for a particular request.

The apparatus and method described herein may advantageously expedite communication between devices in a computer system. By using the delay value received with a response from a target device, a sending device may more effectively time the resending of a request to more closely correspond with the resolution of a temporarily unavailable condition at the target device. As a result, network traffic and latencies associated with the processing of a request may advantageously be reduced. In addition, the apparatus and method may advantageously allow a target device to determine an appropriate time to retry the resending of a request, thereby allowing a target-independent retry policy at a sending device.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

Fig. 1 is a block diagram illustrating one embodiment of devices configured to communicate according to a communications protocol.

Fig. 2 is a block diagram illustrating one embodiment of a computer system.

Fig. 3 is a block diagram illustrating one embodiment of a computer system.

Fig. 4 is a block diagram illustrating one embodiment of a computer system.

Fig. 5 is a flow chart illustrating a method for enhancing communication in between devices.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

### **DETAILED DESCRIPTION OF AN EMBODIMENT**

Turning now to Fig. 1, a block diagram illustrating one embodiment of devices configured to communicate according to a communications protocol is shown. Other embodiments are possible and contemplated. Fig. 1 depicts sending device 110 coupled to target device 120 using communications medium 100. Communications medium 100 may comprise one or more of the communications networks shown in Fig. 2, Fig. 3, and Fig. 4. Sending device 110 and target device 120 can be configured to exchange packets of information or other suitable forms of information according to a communications protocol.

In the embodiment of Fig. 1, sending device 110 and target device 120 can be configured to exchange requests 112 and responses 122 with one another. For example, sending device 110 can be configured to convey a request 112 to target device 120. Target device 120 can be configured to convey response 122 in response to receiving or processing request 112 from device 110. Response 122 conveyed from target device 120 may comprise an acknowledgment (ACK) or a negative acknowledgment (NAK) according to a communications protocol employed by the devices.

At certain times, target device 120 may be temporarily unable to process a request from sending device 110. These periods may be referred to as “temporarily unavailable conditions” and may occur when target device 120 is handling another operation that temporarily prevents the processing of a request from sending device 110 that encounters the temporarily unavailable

condition. Such operations may include a temporarily loss of system resources (e.g., a dynamic reconfiguration of a node), a temporary lack of processing resources on the target device, or a lack of a valid virtual to physical address translation in cases where the contents of the request are to be written in the virtual address space of the target device's node. In response to detecting a temporarily unavailable condition, target device 120 can be configured to convey a negative acknowledgment (NAK) or other type of response to sending device 110. The NAK can indicate to sending device 110 that target device 120 is temporarily unable to process the request received from sending device 110. In certain embodiments, target device 120 can be configured to convey different types of NAKs depending on the type of temporarily unavailable condition detected.

The NAK can include a delay value that can be used by sending device 110 as a hint for determining how long to delay the resending of its request. Using the delay value, sending device 110 may advantageously resend its request at a time when target device 120 may be able to process the request, i.e., after sufficient time to allow the temporarily unavailable condition to be cleared at target device 120. In certain configurations or for certain types of temporarily unavailable conditions, sending device 110 may be configured to ignore the delay value and independently determine when to resend its request.

In one embodiment, target device 120 can be configured to generate a delay value according to the type of operation that is causing a temporarily unavailable condition. In this manner, different delay values can be generated for different types of operations as the different types of operations may vary widely as to the amount of time necessary for target device 120 to clear the temporarily unavailable condition. Target device 120 can generate delay values according to a set value for each type of operation, a programmed value for each type of operation, or a dynamically calculated value for each type of operation. Target device 120 may be configured to store historical data from previous temporarily unavailable conditions and may calculate delay values from this data. Target device 120 may also keep track of the number of outstanding responses it has sent for a particular temporarily unavailable condition. In doing so, target device 120 can convey delay values that indicate longer and longer delay periods as the number of outstanding responses increases. The delay value may be encoded to minimize the

size and/or number of packets needed for the NAK. In one particular embodiment, the delay value can be encoded according to an exponential encoding in order to cover numerous orders of magnitude range.

5 In response to receiving a NAK that includes a delay value from target device 120, sending device 110 can use the delay value to determine when to resend its request. If the delay value is sent in an encoded format, sending device 110 can decode the delay value in order to determine when to resend the request. By using the delay value, sending device 110 may resend the request at a time where target device 120 will more likely be able to process the request  
10 without unnecessarily delaying the resending of the request. In this manner, overall traffic between sending device 110 and target device 120 may be reduced as sending device 110 may reduce the number of times it resends the request (also resulting in a decrease in the number of NAKs sent by target device 120).

15 In certain embodiments, a policy layer can determine a retry limit for a particular request sent by sending device 110. In response to sending device 110 resending its request in excess of the retry limit, the policy layer can be configured to detect an error and can initiate an error recovery mechanism based on the type of NAK most recently received from target device 120. In this manner, the type of NAK can allow for different error recovery mechanisms based on  
20 different types of temporarily unavailable conditions at target device 120. In other embodiments, the policy layer can be configured to detect an error and can initiate an error recovery mechanism based on the delay value corresponding to the most recently received NAK from target device 120.

25 Turning now to Fig. 2, a block diagram illustrating one embodiment of a computer system is shown. Other embodiments are possible and contemplated. Fig. 2 depicts devices 220a, 220b, 220c, 220d, 220e, and 220f coupled to switching network 210. Other embodiments may include any number of devices coupled to switching network 210.

Devices 220a, 220b, 220c, 220d, 220e, and 220f can be configured to communicate with one another through switching network 210 using a communications protocol. Switching network 210 can be configured to receive a request from one of devices 220a, 220b, 220c, 220d, 220e, or 220f and route it to the appropriate device 220a, 220b, 220c, 220d, 220e, or 220f.

5 Similarly, switching network 210 can be configured to receive a response to the request from one of devices 220a, 220b, 220c, 220d, 220e, or 220f and route it to the appropriate device 220a, 220b, 220c, 220d, 220e, or 220f. Devices 220a, 220b, 220c, 220d, 220e, and 220f can be configured to use delay values as described above in Fig. 1.

10 Turning now to Fig. 3, a block diagram illustrating one embodiment of a computer system is shown. Other embodiments are possible and contemplated. Fig. 3 depicts device 310a coupled to device 310b, device 310b coupled to device 310c, device 310c coupled to device 310d, device 310d coupled to device 310e, device 310e coupled to device 310f, and device 310f coupled to device 310a in an arbitrated loop. Other embodiments may include any number of  
15 devices coupled in an arbitrated loop configuration.

Devices 310a, 310b, 310c, 310d, 310e, and 310f can be configured to communicate with one another through the arbitrated loop using a communications protocol. The devices can send and receive requests and responses from the arbitrated loop and can be configured to use delay  
20 values as described above in Fig. 1.

Turning now to Fig. 4, a block diagram illustrating one embodiment of a computer system is shown. Other embodiments are possible and contemplated. Fig. 4 depicts devices 410a, 410b, 410c, 410d, 410e, and 410f coupled to shared bus 420. Other embodiments may  
25 include any number of devices coupled to shared bus 420.

Devices 410a, 410b, 410c, 410d, 410e, and 410f can be configured to communicate with one another across shared bus 420 using a communications protocol. The devices can send and



receive requests and responses from shared bus 420 and can be configured to use delay values as described above in Fig. 1.

Although Fig. 2, Fig. 3, and Fig. 4 illustrate embodiments of configurations for communication between the devices, other configurations and communications media are possible and contemplated.

Turning now to Fig. 5, a flow chart illustrating a method for enhancing communication in between devices is shown. Variations of the method are possible and contemplated. In Fig. 5, a first device can convey a request to a second device as illustrated in block 502. A second device can receive the request as illustrated in block 504. Block 506 illustrates determining whether the second device is temporarily unavailable. If the second device is not temporarily unavailable, then the second device may convey an acknowledgement (ACK) to the first device as illustrated in block 508. If the second device is temporarily unavailable, then the second device can determine a delay value as illustrated in block 510. The second device can convey a NAK that includes the delay value to the first device as illustrated in block 512. Block 514 illustrates determining whether a retry limit has been exceeded. If the retry limit has not been exceeded, then the first device can re-convey the request at a later time according to the delay value as illustrated in block 516. The method can then resume at block 504 as indicated. If the retry limit has been exceeded, then an error recovery mechanism can be initiated according to a type of the NAK as illustrated in block 518.

Although the embodiments above have been described in considerable detail, other versions are possible. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.